

**MAÎTRISE DE PROGICIELS
DE GESTION DE BASES DE DONNÉES
ET DE TRAITEMENT DE TEXTE
Compte rendu d'un stage à l'usage
des professeurs de sciences sociales.**

Bernard HAMM, Évelyne LAVOISIER

Ce stage a été programmé A la demande et avec la collaboration d'un groupe d'enseignants de sciences sociales. Les stagiaires ont été en priorité des enseignants de sciences sociales, de secrétariat, d'EMT ayant pour la plupart quelques notions de gestion automatique de fichiers et de traitement de texte.

Du point de vue administratif, le stage a été proposé par l'EPI dans le cadre des stages d'été IPT et s'est tenu au Lycée Georges de la Tour à Metz.

1 - OBJECTIFS DU STAGE

L'informatique est explicitement aux programmes de certaines sections techniques, en particulier des techniciens en secrétariat médico-social dans le cadre des techniques professionnelles.

C'est ainsi qu'il est précisé dans les programmes :

" 1. Présentation et diffusion de l'information.

...

1.2 Dactylo.

Commentaire

" il est indispensable que les élèves puissent connaître et ai possible utiliser les machines électriques, électro-niques, de traitement de texte..."

.....

3. Traitement de l'information.

.....

3.3 Traitement automatique des informations. Initiation à l'informatique. Principales applications de l'informatique dans les secteurs médical et social.

.....

C'est pour répondre à ces besoins que ce stage a été organisé et centré sur les problèmes de gestion de bases de données, avec une brève initiation au traitement de texte.

L'initiation aux SGDB a été faite à l'aide du progiciel dBase essentiellement pour deux raisons

- il s'agit d'un logiciel largement utilisé dans les milieux professionnels,
- son langage de programmation permet d'élaborer rapidement des applications proches de celles rencontrées dans les mêmes milieux.

2- DÉROULEMENT DU STAGE

L'hétérogénéité des connaissances des stagiaires, tant du point de vue matériel que logiciel, a nécessité une mise à niveau axée sur la maîtrise de la machine, l'utilisation du système d'exploitation et la manipulation d'un traitement de texte.

Un cours (environ 6h) a été proposé à l'ensemble des stagiaires : il a consisté en l'élaboration d'une application de publipostage (voir ci-dessous). Toutes les instructions et les techniques de programmation ont été introduites et commentées lors de leur apparition dans ce programme. Les stagiaires n'ont pas saisi ce programme sur leur machine, mais, utilisant celui-ci comme canevas et comme référence, ils ont réalisé leur propre projet.

Le reste du temps a été consacré à l'analyse et à la réalisation de projets pédagogiques par groupe de deux ou trois stagiaires. Lorsque le besoin s'en faisait sentir, les stagiaires les moins expérimentés ont bénéficié de compléments théoriques et pratiques.

Soulignons que la démarche pédagogique retenue implique la disponibilité constante du matériel : il est important que chaque notion acquise puisse immédiatement être testée puis intégrée à l'application qui se construit ainsi progressivement.

3 - CONNAISSANCES ACQUISES AU COURS DU STAGE

L'utilisation de dBase nécessite la maîtrise de quelques commandes MS-DOS (format, dir, type, copy).

Parmi l'ensemble des possibilités offertes par dBase nous avons retenu :

3.1 - Commandes dBase

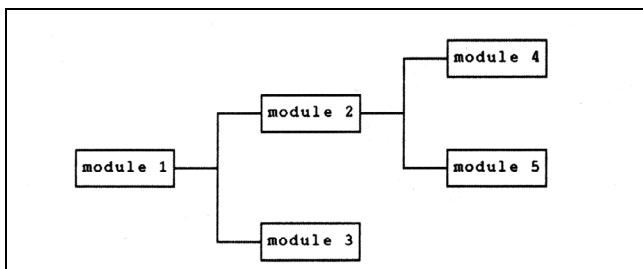
dBase est un langage qui, très souvent, n'est utilisé qu'en mode direct. Voulant insister avant tout sur l'aspect programmation, nous nous sommes limités à certaines commandes essentielles :

- CREATE : création de la structure d'un fichier.
- MODIFY STRUCTURE : modification de la structure d'un fichier.
- MODIFY COMMAND : mise en oeuvre de l'éditeur de textes de dBase permettant la création des programmes.
- DELETE : destruction logique d'une fiche.
- PACK : destruction physique des fiches éliminées par delete.
- QUIT : permet de quitter dBase. Cette commande est indispensable si on veut être sûr de bien récupérer les fichiers (vidage de la mémoire tampon).

3.2 - Programme dBase : structure générale

Un programme dBase est formé de modules dont l'exécution est déclenchée par l'instruction DO < nom du module >. Chaque module est sauvé sur disquette dans un fichier texte ayant pour suffixe PRG. Chaque module est composé de lignes d'instruction, le séparateur d'instruction étant le caractère "

Afin de mieux contrôler le bon fonctionnement d'un programme, on a tout intérêt à donner à celui-ci une structure d'arbre et de représenter celle-ci à l'aide d'un schéma du type suivant.



Lors de la création du programme, on a tout intérêt à établir, pour chaque module, la liste des variables utilisées.

3.3 - Type d'objets manipulés par dBase

a - Les variables mémoires

Ces variables sont créées par programme. Leur type (chaîne, numérique ou booléen) est décidé lors de leur première affectation.

Affectation

STORE <expression> TO <identificateur>

Opérateurs numériques

+ . - * . / -

Opérateurs chaîne

! () . conversion minuscule en majuscule.

+ : concaténation

b - Les fichiers

La structure des fichiers est créée en mode direct. Lorsque le fichier est ouvert, les divers champs d'une fiche sont manipulés comme des variables. Nous n'avons pas jugé nécessaire, dans ce stage d'insister sur les notions d'index et de tri. Liste des instructions étudiées :

- Ouverture d'un fichier : USE <nom du fichier>
- Adjonction d'une fiche vide : APPEND BLANK
- Remplacement du contenu d'un champ par le contenu d'une variable mémoire : REPLACE <nom champ> WITH <nom var>
- Destruction d'une fiche : DELETE
- Booléen de fin de fichier : EOF
- Passage d'une fiche à la suivante : SKIP

Afin d'éviter toute confusion entre les variables mémoires et les champs d'une fiche, nous avons proposé de commencer tous les identificateurs de variable mémoire par la lettre "V". De plus, pour une plus grande lisibilité des programmes, nous avons suggéré l'emploi systématique des minuscules à l'exception des identificateurs.

3.4 - Gestion des entrées et des sorties

Une des grandes originalités de dBase est de permettre une bonne gestion de l'écran. Lors de l'écriture des programmes, nous n'avons pas utilisé l'instruction INPUT lui préférant l'usage systématique de

```
@ <n°l, n°c> GET <identificateur> PICTURE <format> READ
```

Ces instructions permettent de familiariser très vite les débutants avec le contrôle des entrées.

Par contre, lors de l' édition de fiches, il est préférable d'utiliser " ?" (print) à l'instruction SAY afin de pouvoir mettre en oeuvre plus simplement une sortie sur imprimante.

Le choix de l'imprimante comme périphérique de sortie se fait à l'aide des instructions

```
SET PRINT ON
<textes à imprimer>
SET PRINT OFF
```

3.5 - Structures de contrôle

a - Structure de cas

Syntaxe :

```
DO CASE
    CASE <condition 1>
    [ bloc instruction]
    CASE <condition 2>
    ...
    ...
    CASE <condition n>
    [ bloc instruction]
ENDCASE
```

Cette structure permet de répondre à toutes les situations. Pour des raisons de facilité de modification de programmes nous n'avons pas introduit la structure "IF" "ENDIF".

b - Structure itérative

Syntaxe :

```
DO WHILE <condition>
    [bloc instruction]
ENDDO
```

Cette structure a surtout été employée lors de la création de menu et lors du parcours séquentiel des fichiers. Deux erreurs sont à éviter :

- oubli de l'initialisation des variables utilisées dans la condition (en particulier l'oubli de l'ouverture d'un fichier avant l'emploi de EOF)
- oubli du saut de fiche (SKIP) .

4 - PROJET TYPE RÉALISÉ : PROGRAMME DE PUBLIPOSTAGE

Ce projet a permis, lors du cours, de présenter les techniques de base d'analyse et de programmation en dBase. Il nous a semblé important de présenter ici le dossier d'analyse et de programmation développé lors du stage. Pour des raisons de place, ce dossier ne contient ni mode d'emploi, ni trace d'exécution.

4.1 Cahier des charges

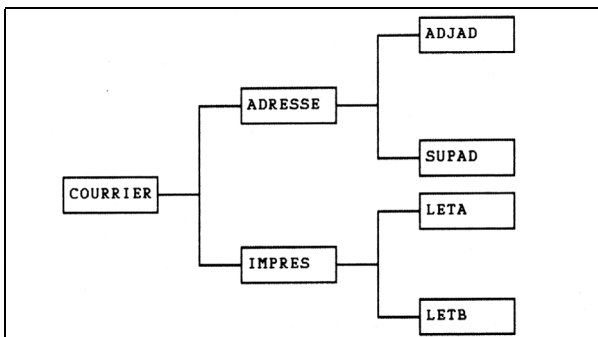
Ce programme de publipostage doit permettre

- 1 - de gérer un fichier CLIENT contenant le nom et l'adresse des clients d'une entreprise.
- 2 - d'éditer deux types de lettres adressées aux clients de cette entreprise.

Ce programme a été conçu de façon minimale : la gestion du fichier se fera par adjonction et suppression de fiches et l'édition des lettres se fera sur l'ensemble du fichier CLIENT.

4.2 Structure du programme

a - Représentation graphique des modules utilisés



b - Lexique du programme

COURRIER	Programme principal. Ce programme permet un choix entre le travail sur le fichier client et l'édition de lettres		
	WREP1	Chaîne	Choix utilisateur
ADRESSE	Module appelé par COURRIER Permet un choix entre l'adjonction de fiches et la suppression de fiches dans CLIENT		
	WREP2	Chaîne	Choix utilisateur
IMPRES	Module appelé par COURRIER Permet un choix entre deux types de lettres		
	WREP3	Chaîne	Choix utilisateur
ADJAD	Module appelé par ADRESSE Permet l'adjonction d'une fiche au fichier CLIENT		
	CLIENT	Fichier	Fichier utilisé par le pgm
SUPAD	Module appelé par ADRESSE Permet la suppression d'une fiche au fichier CLIENT		
	CLIENT	Fichier	Fichier utilisé par le pgm
	WNOM	Chaîne	Nom du client
	WPRENOM	Chaîne	Prénom du client
LETA	Module appelé par IMPRES Permet l'impression du premier type de lettre		
	CLIENT	Fichier	Fichier utilisé par le pgm
LETB	Module appelé par IMPRES Permet l'impression du deuxième type de lettre		
	CLIENT	Fichier	Fichier utilisé par le pgm

Remarque : Sous chaque module, on indique la liste des variables et des fichiers utilisés.

c - Structure du fichier CLIENT

CLIENT	Fichier dBase		
NOM	Chaîne	20	Nom du client
PRENOM	Chaîne	20	Prénom du client
RUE	Chaîne	30	Rue du client
CODPOST	Chaîne	5	Code postal de la ville
VILLE	Chaîne	20	Nom de la ville

4.3 Liste des instructions du programme

4.31 Programme principal

```
***COURRIER.PRG*** ***
```

```
PROGRAMME PRINCIPAL
```

```
set talk off
```

```
store "1" to WREP1
```

```
do ville LTREP1<>"0"
```

```
    erase
```

```
    text
```

COURRIER

VOULEZ-VOUS ?

0 ARRETER

1. GERER LE FICHER ADRESSES

2. IMPRIMER

VOTRE CHOIX

```
endt ext
```

```
@ 10,33 get WREP1 picture "9"
```

```
read
```

```
do case
```

```
    case IJREPI="1"
```

```
        do ADRESSE
```

```
    case WREP1="2"
```

```
        do IMPRES
```

```
endcase
```

```
enddo
```

```
set talk on
```

Remarques :

- La structure de menu de ce programme est très fréquente en dBase. On la retrouve dans les deux modules suivants.
- Pour accélérer la programmation ce module peut être copié soit sous MS-DOS, soit sous traitement de texte.

4.32 Module ADRESSE

```
****ADRESSE.SPG1****
```

```
* Sous programme de gestion de fichier d'adresses appelé par COURRIER store "1" to WREP2
```

```
do while WREP2<>"0"
```

```
    erase
```

```
    text
```

ADRESSE . COURRIER

Gestion du fichier d'adresses


```

        Voulez-vous ?
        0 - Arrêter
        1 - Ajouter une adresse
        2 - Supprimer une adresse
        Votre choix
    endtext
    @ 9,33 get WREP2 picture "9"
    read
    do case
        case WREP2="1"
            do ADJAD
        case WREP2="2"
            do SUPAD
    endcase
enddo
return

```

4.33 Module IMPRES

```

****IMPRES.SPG2****
*Sous programme d'impression appelé par COURRIER store "1" to WREP3
do while WREP3<>"0"
    erase
    text
                                IMPRES.COURRIER

        Voulez-vous :
        0 - Arrêter
        1 - Imprimer une convocation
        2 - Imprimer une lettre de rappel
        VOTRE CHOIX
    endtext
    @ 9,33 get WREP3 picture "9" read,
    do case
        case WREP3="1"
            do LETA
        case WREP3="2"
            do LETB
    endcase
enddo
return

```

4.34 Module ADJAD

```

****ADJAD.SSPG11****
*Sous programme d'adjonction d'adresses appelé par ADRESSE use CLIENT
append blank erase
    text

```

ADJONCTION DE FICHE

```

      NOM :                Prénom :
      Rue :
      Code postal :       VILLE :
endtext
@ 4,24 get NOM           picture "!!!!!!!!!!!!!!!!!!!!!!"
@ 4,52 get PRENOM       picture "!AAAAAAAAAAAAAAAAAAAAA"
@ 6,24 get RUE          picture "!!!!!!!!!!!!!!!!!!!!!!"
@ 8,32 get CODPOST      picture "99999"
@ 8,46 get VILLE        picture "!!!!!!!!!!!!!!!!!!!!!!"
read
return

```

Remarques : L'instruction [append blank] crée une fiche vide en fin de fichier. Ce module modifie le contenu de cette fiche.

4.35 Module SUPAD

```

*****SUPAD.SSPG12*****
*****Sous-programme de suppression d'adresse appelé par ADRESSE*****
use client
erase
text
                                     SUPPRESSION DE FICHES
      La fiche de quel client voulez-vous supprimer ?
      Donnez son nom :
                son prénom :
endtext
store "                " to WNOM
store "                " to WPRENOM
@ 4,28 get WNOM        picture "!!!!!!!!!!!!!!!!!!!!!!"
@ 5,28 get WPRENOM     picture "!!!!!!!!!!!!!!!!!!!!!!"
read
do while not. EOF
  do case
    case WNOM=NOM and. WPRENOM=PRENOM
      delete
    encase
  skip
enddo
return

```

Remarques :

- EOF est une variable booléenne qui est vraie lorsque le pointeur de fiche est sur la dernière fiche.
- la boucle de recherche de fiche utilisée ici est, par sa simplicité, adaptable à de nombreux algorithmes. Cependant, elle est très lente sur de grands fichiers.

4.36 Module LETA

```

*****LETA.SSPG21*****
*Sous programme d'impression de convocation appelé par IMPRES use
CLIENT
set print on
do while not. EOF
    erase
    ? NOM, PRÉNOM ? RUE
    ? CODEPOST, VILLE
    text
                                Metz
                                le
                                CONVOCATION
    endtext
    skip
enddo
set print off
return

```

Remarques : Lors de la mise au point de ce module, on a intérêt à remplacer [set print on) par [* set print on] - LETB a rigoureusement la même structure.

5 - PROJETS RÉALISÉS PAR LES STAGIAIRES

Partant du projet type ci-dessus, les stagiaires ont réalisé un certain nombre de logiciels utilisant un ou deux fichiers.

Parmi les réalisations, on relève :

- l'adaptation du logiciel de publipostage à l'envoi aux entreprises de la région de lettres de demande de stage.
- un logiciel de facturation d'acte médicaux : le fichier contient les références du patient et de l'acte (type, nombre) et l'impression de lettres était remplacée par celle de factures.
- un logiciel permettant la recherche de l'adresse des intervenants socio-médicaux par secteur géographique, le fichier des intervenants pouvant être mis en relation avec un fichier de familles.

Bernard HAMM, Évelyne LAVOISIER
Professeurs au Lycée G. de la Tour de METZ.